

RECEIVED
CENTRAL FAX CENTER

JUL 05 2006

REMARKS**I. INTRODUCTION**

Claims 1 and 7 have been amended. No new matter has been added. Thus, claims 1-7 remain pending in this application. It is respectfully submitted that based on the above amendments and the following remarks that all of the presently pending claims are in condition for allowance.

II. THE 35 U.S.C. § 112 REJECTIONS SHOULD BE WITHDRAWN

The Examiner has rejected claims 1-7 under 35 U.S.C. § 112 as indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. (See 4/4/06 Office Action, pp. 2-3, ¶ 3). Specifically, the Examiner asserts that claim 7 contains terms that lack antecedent in basis and contains language that is unclear. The Examiner also asserts that claim 1 contains language that is unclear, of which claims 2-5 depend from. Applicant is unsure of the basis for the 35 U.S.C. § 112 rejection of claim 6. Claim 6 is an independent claim and applicant respectfully submits that no terminology in claim 6 lacks antecedent basis or is unclear.

Claim 7 has been amended according to the Examiner's suggestions to distinguish the terms that lack antecedent in basis. Specifically, "the program code segment" on line 5 of claim 7 has been amended to recite "a program code segment." Also, "the privileged processing mode" on line 7 of claim 7 has been amended to recite "a privileged processing mode." Claim 7 has also been amended to clarify the use of the term "software code" as it appears on the amended claim on lines 3 and 5. Specifically, on line 3, "software code" has been amended to "a software code." Also, on line 5, "software code" has been amended to "the software code." Thus, it is respectfully submitted that the Examiner should withdraw the 35 U.S.C. § 112 rejection of claim 7.

Claim 1 has been amended to recite "a software" on line 4 and "the software" on line 8. With the inclusion of these articles and proper antecedent basis, it is clear that the "software"

referred to in lines 4 and 8 are one in the same. Thus, it is respectfully submitted that the Examiner should withdraw the 35 U.S.C. § 112 rejection of claim 1 and all depending claims (*i.e.*, claims 2-5).

III. THE 35 U.S.C. § 101 REJECTIONS SHOULD BE WITHDRAWN

The Examiner has rejected claim 7 under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. (See 4/4/06 Office Action, p. 3, ¶ 5).

As suggested by the Examiner (See Id.), claim 7 has been amended to recite "A computer readable medium encoded with a software application...." With the addition of the Examiner's suggestion in order to make the claim proper statutory subject matter, it is respectfully submitted that the Examiner should withdraw the 35 U.S.C. § 101 rejection of claim 7.

IV. THE 35 U.S.C. § 103(a) REJECTIONS SHOULD BE WITHDRAWN

The Examiner has rejected claims 1-7 under 35 U.S.C. § 103(a) as unpatentable over U.S. Pat. No. 6,175,916 (Ginsberg) in view of U.S. Pat. No. 6,542,919 (Wendorf). (See 4/4/06 Office Action, pp. 4-6).

Ginsberg is directed toward a method of making a call from one process to another. An invalid destination address such as an odd virtual memory address is specified when executing a jump instruction from a first process. The invalid address causes a memory fault, which in turn results in the invocation of a memory fault handler. The memory fault handler is configured to recognize certain specified invalid addresses, and to call different system functions depending on the particular values of the addresses. (See Ginsberg, abstract). Ginsberg also describes a normal function call which is the result of a jump instruction. Code generated by a compiler pushes function arguments onto a memory stack and/or to processor registers and then issues a call instruction. (See Id., col. 8, ll. 12-15).

Wendorf is directed toward an operating system that provides protection domain support

to be compatible with certain threads that were written without regard for protection domains. These threads are defined as “well behaved” threads which are threads that obtain all their memory allocations from the operating system. (See Wendorf, abstract). One main feature of the method in Wendorf is to initially determine if the thread is in the system domain. (See Figs. 2, 3, and 5, and col. 7, ll. 1-46). Specifically, as per Fig. 2, a conditional branch point tests to determine if the thread whose execution is being initiated is in the protection domain of the operating system. (See *Id.*, at ll. 4-6).

Claim 1 recites “when the current processing mode is a privileged processing mode” and “when the current processing mode is an unprivileged processing mode.” The Examiner asserts that Ginsberg discloses this recitation of claim 1. (See 4/4/06 Office Action, p. 4, ¶ 8). Applicants respectfully disagree. Ginsberg performs the memory fault handler despite the state of the current processing mode. The normal function call described in Ginsberg is also performed despite the state of the current processing mode. In fact, there is no disclosure within Ginsberg that relates to the state of the current processing mode. This is further evidenced by the Examiner’s statement that Ginsberg is silent with respect with determining a current processing mode of an executing software function, to be discussed below. (See *Id.*, p. 4, ¶ 9). Those of skill in the art would understand that without a first determination, the state of the current processing mode is inaccessible.

In addition, the Examiner has correctly stated that Ginsberg fails to disclose “determining a current processing mode of an executing software function,” as recited in claim 1. (See *Id.*). The Examiner has attempted to cure this deficiency with Wendorf. (See *Id.*, p. 5, ¶ 10). However, Applicants respectfully disagree that Wendorf cures this deficiency.

As discussed above, Wendorf makes a determination of whether the thread is in the protection domain of the operating system. (See Wendorf, Figs. 2 (step 203), 3 (step 303), and 5 (step 503)). That is, determination of the presence or absence of the thread is not a determination of a current processing mode. In the present invention, the examination of the “current processing mode” is used to decide whether an executing software function should be allowed to access a system resource directly or indirectly and not whether it exists in the protection domain

or not. The software functions of the present invention are presumed to already exist. Thus, the present invention need only to allow the computing system to decide the type of access that the software functions will obtain – direct or indirect.

Thus, it is respectfully submitted that neither Ginsberg nor Wendorf, either alone or in combination, discloses or suggests “determining a current processing mode of an executing software function,” as recited in claim 1. Accordingly, it is respectfully submitted that the Examiner should withdraw the 35 U.S.C. § 103(a) rejection of claim 1. Because claims 2-5 depend from and, therefore, include all the limitations of claim 1, it is also respectfully submitted that these claims are allowable for at least the reasons stated above.

Independent claim 6 recites “identify a current processing mode of the program code segment.” Independent claim 7 recites “identify a current processing mode of a program code segment.” Thus, Applicants respectfully submit that these claims are allowable for at least the same reasons stated above with reference to claim 1, and the Examiner should withdraw the 35 U.S.C. § 103(a) rejections for these claims.

RECEIVED
CENTRAL FAX CENTER

JUL 05 2006

CONCLUSION

In view of the above remarks, it is respectfully submitted that all the presently pending claims are in condition for allowance. All issues raised by the Examiner having been addressed, an early and favorable action on the merits is earnestly solicited.

Dated: July 5, 2006

Respectfully submitted,

By: 
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, New York 10038
Tel: (212) 619-6000
Fax: (212) 619-0276